MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

Location-Based Tracking of Moving Obstacles from a Mobile Robot

Thatsaphan Suwannathat, Jun-ichi Imai and Masahide Kaneko
The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585, JAPAN
touch@radish.ee.uec.ac.jp, imai@ee.uec.ac.jp and kaneko@ee.uec.ac.jp

Abstract

Tracking moving obstacles from a moving platform is a useful skill for the coming generation of mobile robot. The methods used in existing moving objects tracking that operated from fixed platform and fixed background, are not applicable. Instead, we propose a system that detects unexpected moving obstacles that appear in the path of mobile robot. The system is designed to track and determine the location of the obstacles with the use of calibrated stereo camera mounted on the top of robot. This method is implemented in the domain of moving objects tracking from a mobile platform using a novel combination of motion segmentation, using spatiotemporal segmentation from the left image sequence, and the 3-D information which is calculated from a stereo image sequence. Position estimation of the moving objects is performed by using Kalman filter on the plan-view map. The system was implemented and tested on an Omni-directional robot at our laboratory. The results show that we are able to reliably detect and track moving objects in natural environments approximately 3×8 meters in front of mobile robot. Our system is able to detect the moving objects at 13 Hz for a 320×240 pixels stereo image sequence using a standard laptop computer.

1. Introduction

Motion detection from a moving observer has been a very important technique for computer vision applications. One of its most important tasks is to detect the moving obstacles like moving humans while the observer itself is running. Methods of differencing image with the clear background or between adjacent frames are well used for the motion detection. But when the observer is also moving, which leads to the result of continuously changing background scene in the perspective projection image, it becomes more difficult to detect moving objects by differencing methods.

The notable algorithms for detecting moving objects from a moving platform using only a vision sensor can be grouped into 3 classes, namely: 1) methods using optical flow, 2) methods using qualitative estimates of motion, 3) methods using stereo-based technique.

The first algorithms, first compute the optical flow from the whole image and then use the optical flow information to analyze scenes obtained by a moving observer [1]. A disadvantage of this method is the difficulty in computing the optical flow with an acceptable noise.

The second algorithms use image transformation and qualitative analysis of the motion of scene point to detect and segment the moving obstacles [2]. These methods are typically more effective in the object segmentation method, but additional process is required to get information about the object's motion. The focus of expansion (FOE) is an important role of this algorithm. However, determining the FOE in real scenes is a nontrivial problem.

The third algorithms use stereo-based technique to perform the detection of moving objects. In [3], this system proposed a pedestrian detection method based on the use of stereo-based segmentation and neural-based recognition. Detecting schemes using only shape information tend to give large false detection rates because there are often many objects that can have a profile similar to that of a human. In [4], a "*v-disparity*" algorithm in stereovision was proposed to detect position of the obstacle. This method fails when there are many objects in the scene.

In this paper we consider the problem of simultaneously tracking unexpected moving objects that appear in the path of moving robot (when the robot is moving). The moving objects in this paper are in the form of moving humans. The aims of the system are: 1) to realize the task of moving objects detection from the robot with the aim of a real-time effective implementation; 2) to determine the location of each moving objects within the

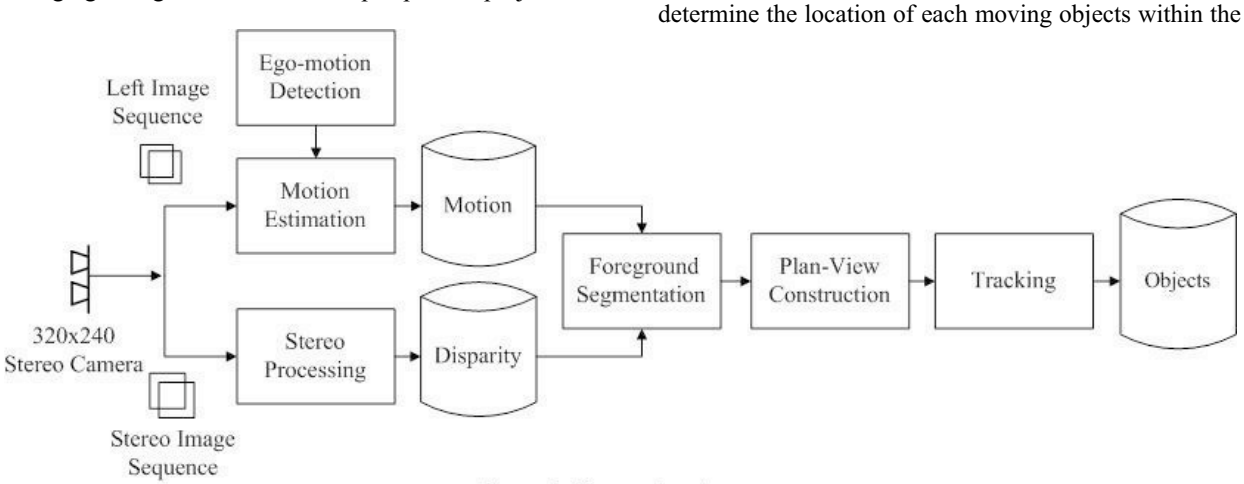


Figure 1. Proposed system.

environment; 3) to maintain a correct association between moving objects and tracks over time.

To combat all the above problems, we introduce a new combination of spatiotemporal segmentation based on location detection that better preserves object location information than prior approaches. The system is able to locate and track multiple moving obstacles that appear in the scene, by using three different representations of the information coming from a stereo vision sensor: the left intensity image, the disparity image and the 3-D location of measured points.

The rest of the paper is organized as follows. Section 2 describes proposed system overviews. The ego-motion and motion estimation using spatiotemporal segmentation method are introduced in Sec.3. Section 4 and 5, disparity motion detection and plan-view construction are presented. Section 6 describes the details of moving objects tracking. Experimental results are discussed in Sec.7. Finally, in Sec.8, conclusion and further research topics are presented.

2. System Overviews

Our proposed system is depicted in Fig. 1. To extract moving objects, the left image sequence from a stereo camera which is mounted on the top of a mobile robot, is fed into the *Motion estimation module*. Subsequently, a stream of stereo images, captured by a calibrated stereo-vision device, is processed by the *Stereo processing module.*

Once we have motion and 3-D information of the scene, these information are then passed to the *Foreground segmentation module*. Then, points in the foreground disparity are projected to the plan-view map in the *Plan-view construction module*. In this module, by considering the blob of pixels in plan-view map, we are able to determine the locations corresponding to each object.

Finally, the set of segmented information for each person from *Plan-view construction module* are then passed to the *Tracking module*, which maintains and tracks a set of tracked objects.

3. Motion Estimation

3.1 Ego-Motion Detection

Ego-motion is calculated according to [5]. First the position and orientation change of camera between two successive frames are computed. Then, according to Eq. (1) a predicted image is calculated and matched with the actually grabbed image.

$$x_t = f\frac{x_{t-1} + \alpha \sin\theta y_{t-1} + f\alpha\cos\theta}{-\alpha\cos\theta x_{t-1} + \gamma y_{t-1} + f}$$
$$y_t = f\frac{-\alpha\sin\theta x_{t-1} + y_{t-1} + f\gamma}{-\alpha\cos\theta x_{t-1} + \gamma y_{t-1} + f} \quad (1)$$

where f is the focal length, θ is the inclination of the camera which is acquired from mobile robot, α and γ are pan and tilt, respectively. With the knowledge of f, θ, α and γ for every pixel position in the previous image (x_{t-1},y_{t-1}), we can calculate the position of the corresponding pixel (x_t, y_t) in the current pixel.

3.2 Motion Detection using Adjacent Mean Difference

In order to remove moving pixels because of observer motion, our new method called adjacent mean difference based on spatiotemporal segmentation is proposed [6].

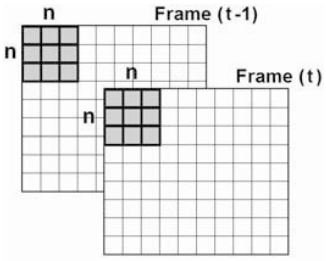


Figure 2. The steps of motion estimation.

By determining the ego-motion in the previous section, the size of window mask can be formulated as follows:

$$n = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \quad (2)$$

Let S_{xy} represent the set of coordinates in a rectangular window of size $n \times n$, centered at point (x,y) as shown in Fig. 2. The adjacent mean process computes the average value of image $g(s,t)$ in the area defined by S_{xy}. The value of the restored image m at any point (x,y) is simply the adjacent mean computed using pixels in the region defined by S_{xy}. This method can be formulated as follows:

$$m(x,y) = \frac{1}{n \times n}\sum_{(s,t)\in S_{xy}} g(s,t) \quad (3)$$

After the computation of the adjacent mean in each frame, the absolute difference of two consecutive frames (m_i^t *and* m_i^{t-1}) *is* calculated. By determining the suitable threshold T_1, we can label the background and moving model. The result of adjacent mean difference motion detection is shown in Fig. 3(b).

$$M_i(x,y) = \begin{cases} 1, & \text{if } \left|m_i^t(x,y) - m_i^{t-1}(x,y)\right| > T_1 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

4. Foreground Disparity Segmentation

4.1 Disparity Image

In our system, we choose a stereo disparity image for tracking the moving objects. Given a left and right image pair from a stereo camera, we first compute stereo image using the area correlation method described in [7]. The disparity is inversely related to depth according to the formula:

$$d = (x_l' - x_r') = \frac{b \times f}{Z} \quad (5)$$

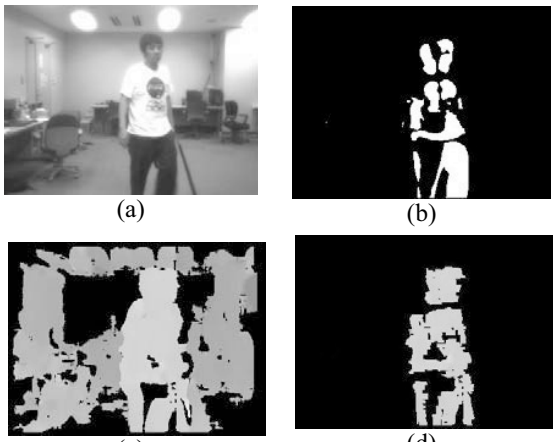
where d is the disparity, b is the stereo camera baseline, f is the camera focal length and Z is the normal distance from the image plane to the object. Figure 3(c) shows the disparity segmentation image.

4.2 Region Growing

A fast region growing segmentation algorithm named "*LabelMinMax*" [8] is carried out to extract regions.

$$Homogeneity: \left(\max_{p \in R} \{ d_t(p) \} - \min_{p \in R} \{ d_t(p) \} \right) \le T_3 \qquad (6)$$

It starts with seed pixel p of the adjacent mean difference image. Region R is the current region which contains the pixel p in the disparity image d_t. The next thing *LabelMinMax* does is to check all the neighboring pixels of region R in the disparity image to find whether region R is still homogenous if neighborhood pixel is inserted into it. A pixel can be added into region R if this insertion does not destroy that region's homogeneity. Gradually, region R expands to all the homogenous territories where no more neighborhood pixels can be added. This process continues until there are no seed pixels left in adjacent mean difference image. The result of region growing is shown in Fig.3(d). In Fig.3(d), it is observed that the foreground disparity pixels are well covered by white regions.



(a) (b) (c) (d)

Figure 3. (a) Original image, (b) Motion mask image, (c) Disparity image, (d) Foreground disparity image.

5. Plan-view Constructions

In many applications it is important to know the 3-D location of tracked objects. We do this by employing a *plan-view* [9]. A plan view assumption allows us to completely model instantaneous foreground information as a 2-D orthographic density projection.

We project (x_j, y_j, d_j) from each point p_j in foreground image into world coordinate (X_w, Y_w, Z_w) using the calibration given by camera, X_w, Y_w are chosen to be orthogonal axes on the ground plane, and Z_w normal to the ground plane. We then compute the plan view volume with;

$$P(X_w, Y_w) = \sum_{\{p_j | X_w = x, Y_w = y, t_w = t\}} 1 \qquad (7)$$

Figure 4 shows the example of plan-view constructions with two moving objects. Figure 4(c) shows 3-D world coordinate plot of foreground images in Fig.4(b). Figure 4(d) shows the plan-view image corresponding to the 3-D plot of Fig.4(c).

6. Tracking of Moving Objects

Our method begins to detect an object by convolving the plan-view image with a square window S_{xy} of width $2 \times W_t$ to find the maximum points. The window size is a parameter with width and height equal to physically average torso width W_t of a human. We use W_t =0.5m. The window size in pixels depends on the plan-view image resolution; our window is about 25 pixels on a side.

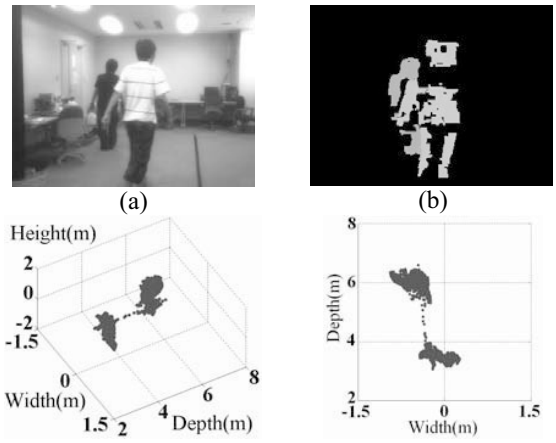
$$f(x, y) = \max_{(X_W, Y_W) \in S_{xy}} \{ P(X_W, Y_W) \} \qquad (8)$$

where $f(x,y)$ is the maximum value of plan-view image $P(X_w, Y_w)$ in the area defined by S_{xy}.

In detecting a new object, if the maximum value $f(x,y)$ is above a threshold T_t, we regard the location as that of a new object. In our tracking system, the threshold T_t is approximately set to the half of average size of a human.

$$T_t = \frac{2 \times W_t}{2} = W_t \qquad (9)$$

After the object has been detected, we delete the plan-view data within a square of width $2 \times W_t$ centered at the location of the window convolution maximum. Then we apply the window convolution to plan-view image again to look for another candidate for new object location. The process continues until the convolution value is below T_t indicating that there is no more location of a new object.



(a) (b) (c) (d)

Figure 4. (a) Original image, (b) Foreground disparity image, (c) 3-D plot of (b), (d) Plan-view project of (c).

7. Experimental Results

For the hardware used in the experiment is the Omni-directional mobile robot. We use the speeds of mobile robot at 25 cm/s to avoid wheel slippage and remove motion blur. To control the robot and process the moving obstacle detection, we have implemented on a standard labtop computer with a 3.2GHz CPU. For stereo camera, we use a Point Grey BumbleBee stereo module at 320x240 resolutions mounted on the top of robot with about 1.4m in height.

The overall system run at 17 Hz when tracking one obstacle, and performance goes down to 13 Hz when tracking two obstacles. This frame rate can be obviously

increased through the use of faster processors and better optimized code, but one should also note that the Triclop's software computation of depth is the most computationally expensive process of our system.

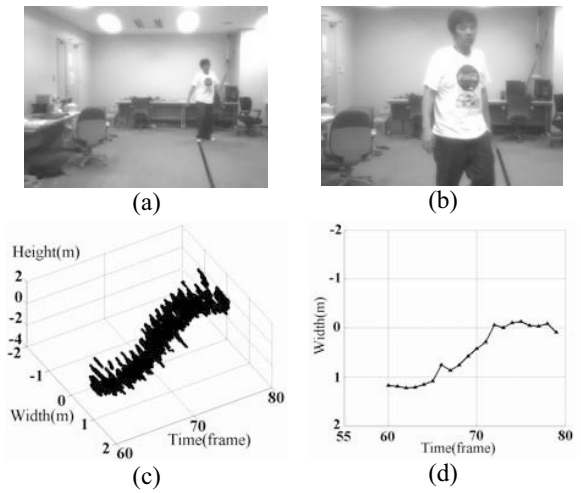


Figure 5. The results of the tracking of single moving object. (a) Start frame, (b) End frame, (c) A plot of foreground density over time, (d) Tracking object using Kalman filtering.

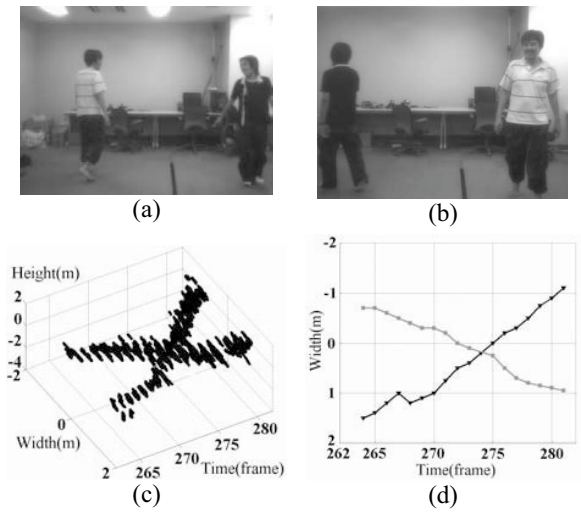


Figure 6. The results of the tracking of two moving objects. (a) Start frame, (b) End frame, (c) A plot of foreground density over time, (d) Tracking object using Kalman filtering.

Figure 5 shows the results of a single moving obstacle tracking. In this sequence, there is a human walking toward the moving robot with a little zigzag pattern.

Figure 6 shows the results of the tracking of two moving obstacles moving cross each other.

From our experiments, the system works well with up to two moving objects in the scene. With more than two moving objects, the frequent occlusions cause enough poor clustering in the stereo module that the system cannot maintain coherent tracks. In the case of more than two moving objects, they contain dozens of majority or complete occlusions of objects by one or more other objects, and contain many close inter-objects interactions of extended duration. This error shows that location-based tracking is not completely reliable in detecting many obstacles. We must supplement the other methods for improving the tracking performance, for example, color-based or template-based tracking.

8. Conclusion

In this paper, we have presented a system to detect the moving objects that appear in the path of a moving robot. The system is designed to track and determine the location of the objects with the use of calibrated stereo camera mounted on the top of robot. A vision-based algorithm used a novel combination of motion segmentation, using spatiotemporal segmentation, and the 3-D information which is acquired from a stereo module. Location estimation of each moving object is maintained by using Kalman filter on the plan-view map. The system was implemented and tested on an Omni-directional mobile robot. The system is able to detect moving obstacles at frame rate of about 15 Hz limited by the speed of the stereo processing in the stereo module. In our experiments, we do not require the demonstrators to wear special clothes. The detection area is approximately 3×8 meters in front of mobile robot.

In the near future we intend to extend the system to track moving obstacles with multiple sensors by using the fusion of sound and vision modalities to achieve a more robust tracking.

References

[1] F. Arnell and L. Petersson, "Fast Object Segmentation from a Moving Camera," Proc. of IEEE Intelligent Vehicles Symposium, pp.136-141, 2005.

[2] R. Okada, Y. Taniguchi, K. Furukawa and K. Onoguchi, "Obstacle Detection Using Projective Invariant and Vanishing Lines," Proc. of the 9th IEEE International Conference on Computer Vision, pp.500-504, 2003.

[3] M. Szarvas, A. Yoshisawa, M. Yamamoto and J. Ogata, "Pedestrian Detection with Convolution Neural Networks," Proc. of IEEE Intelligent Vehicles Symposium, pp.224-229, 2005.

[4] D.M. Gavrila, J. Giebel and S. Munder, "Vision-Based Pedestrian Detection: The PROTECTOR System," Proc. of IEEE Intelligent Vehicles Symposium, pp. 13-18, 2004.

[5] D. Murray and A. Basu, "Motion Tracking with an Active Camera," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 16, No.5, pp.449-459, 1994.

[6] T. Suwannathat and M. Kaneko, "Motion Detection using Omni-Directional Camera Mounted on Moving Platform," Proc. of the 2005 IEICE National Conference, D-12-40, 2005.

[7] K. Konolige, "Small vision systems: hardware and implementation," Proc. of 8th International Symposium on Robotics Research, pp.111-116, 1997.

[8] C. Gu and M. Lee, "Semantic Video Object Tracking Using Region-Based Classification," Proc. of the 2005 IEEE Int. Conf. on Image Processing, pp. 643-647, 1998.

[9] T. Darrell, D. Demirdjian, N. Checka and P.F. FelzensAwalb, "Plan-view Trajectory Estimation with Dense Stereo Background Models," Proc. of 8th Int. Conf. on Computer Vision, pp.628-635, 2001.